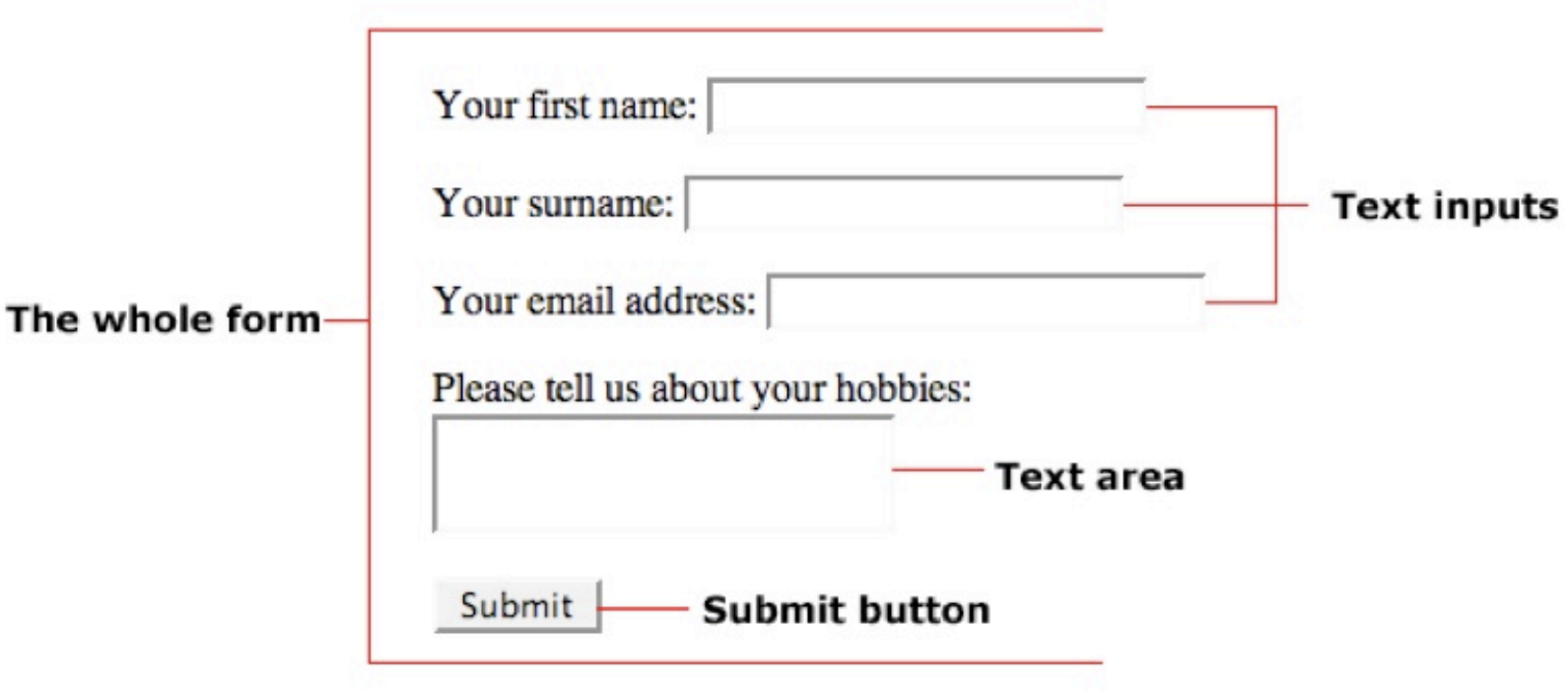
Forms: Interacting with your Audience

SI539 - Lloyd Chapter 7

These slides intended for use with the text book, "Building your Own Web Site the Right Way Using HTML & CSS" by Ian Lloyd, and Available from Sitepoint books.

Forms - Input on the Web

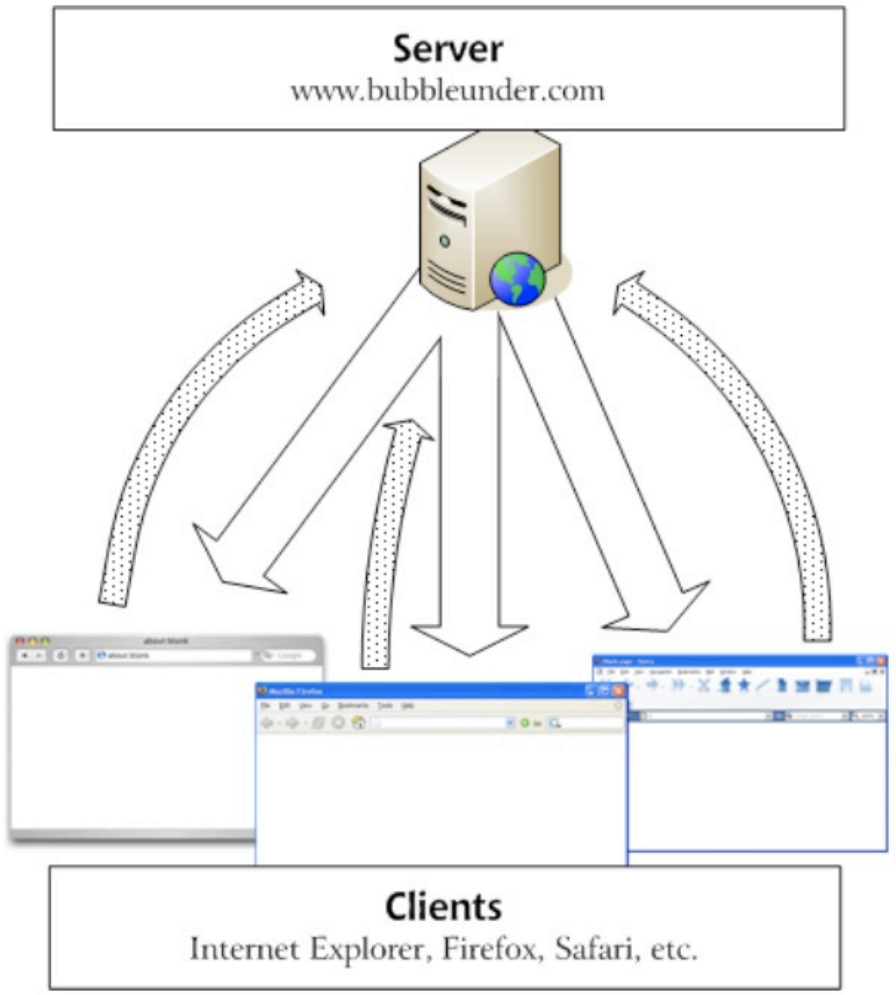


Forms Need Servers

- Forms effectively gather data from the user and "submit" it to a web page on a server
- The earliest form of server-side processing was called GGI - Which stood for Common Gateway Interface
- CGI allows software to "receive" the input parameters and produce the HTML response - rather than simply reading the HTML content from a file

http://en.wikipedia.org/wiki/Common_Gateway_Interface

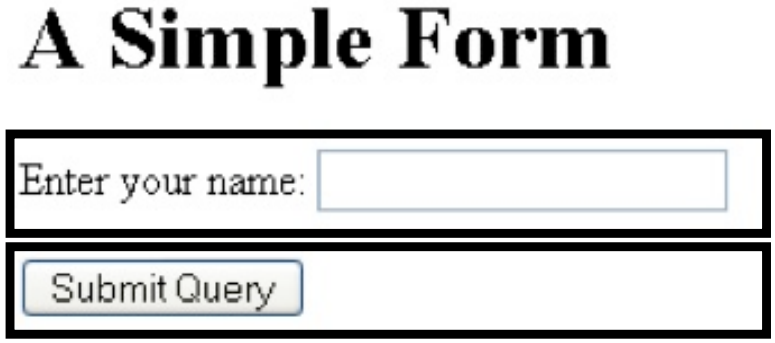
- Typical Server
  - Permanently connected to the network
  - Has static address
- Clients
  - Browsers
  - Many clients at the same time using the server



Using Forms Without Servers

- Submitting form data works without a server - the browser moves from static page to static page
- But the data from the forms is neither saved no is it usable

Forms

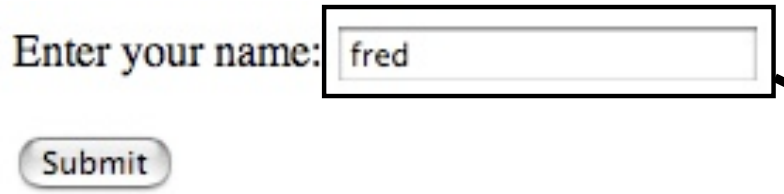



```
<form method="get" action="simpleform.html">
  <p>
    <label for="yourid">Enter your name:</label>
    <input type="text" name="yourname" id="yourid" />
 </p>
  <p><input type="submit" /></p>
</form>
```

Note: <p> tags are used to group labels with their corresponding input field.

Pressing Submit (Get)

- When you fill in a form and press "Submit" the browser packs up the parameters of the form and sends them to the server using the "name=" values as the parameter names and the field contents as the values.
- Then this request returns new HTML which is shown in the browser.



Pressing "Submit"

http://server/.../simpleform.html?yourname=fred

```
<form method="get" action="simpleform.html">
  <p>
    <label for="yourname">Enter your name:</label>
    <input type="text" name="yourname" id="yourid" />
 </p>
  <p><input type="submit" /></p>
</form>
```

Web servers look for the parameters after the "?" in the URL hand hand those parameters to software such as PHP or Rails running in the server.

Get .vs. Post

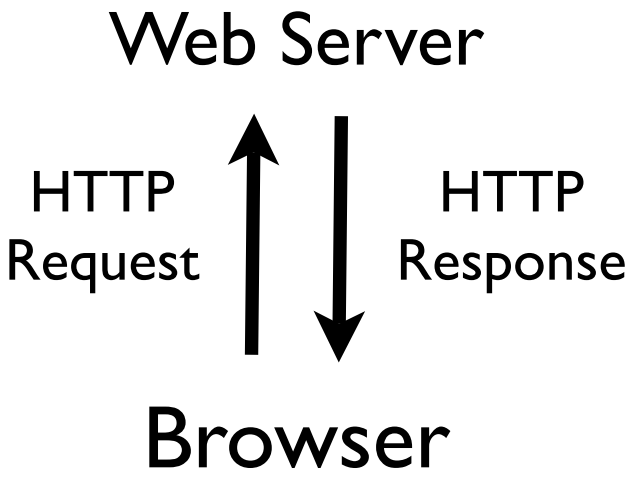
- Two ways the browser can send parameters to the web server
  - GET - Parameters are placed on the URL which is retrieved
  - POST - The URL is retrieved and parameters are appended to the request in the the HTTP connection

<nerdy-stuff>

Getting Data From The Server

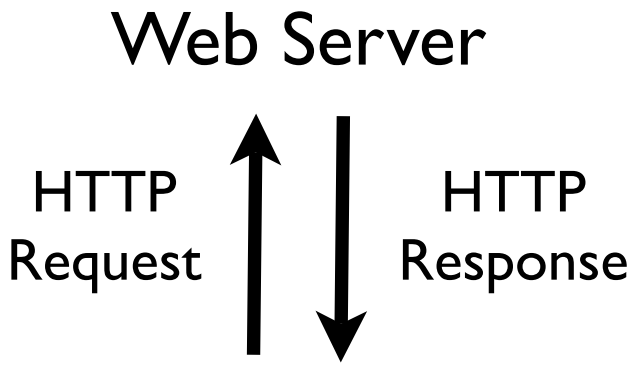
- Each the user clicks on an anchor tag with an href= value to switch to a new page, the browser makes a connection to the web server and issues a "GET" request - to GET the content of the page at the specified URL
- The server returns the HTML document to the Browser which formats and displays the document to the user.

HTTP Request / Response Cycle



http://www.oreilly.com/openbook/cgi/ch04_02.html

HTTP Request / Response Cycle

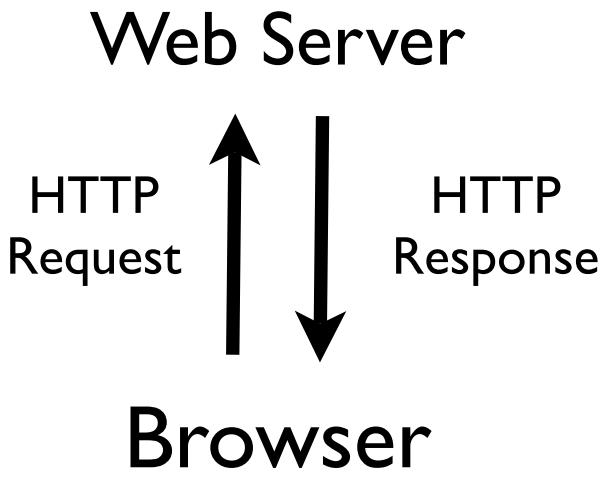


GET /index.html
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14

```
<head> .. </head>
<body>
<h1>Welcome to my
   application</h1>
 ....
</body>
```

http://www.oreilly.com/openbook/cgi/ch04_02.html

“Hacking” HTTP

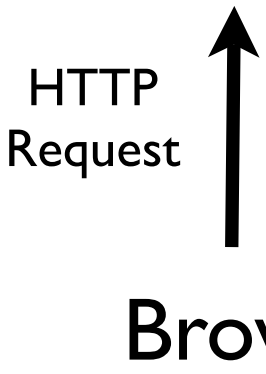



```
Last login: Wed Oct 10 04:20:19 on ttyp2
si-csev-mbp:~ csev$ telnet www.umich.edu 80
Trying 141.211.144.188...
Connected to www.umich.edu.
Escape character is '^]'.
GET /
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
 ....
```

Forms Get .vs. Post

- Two ways the browser can send parameters to the web server
  - GET - Parameters are placed on the URL which is retrieved
  - POST - The URL is retrieved and parameters are appended to the request in the the HTTP connection

Passing Parameters to The Server




```
GET /simpleform.html?yourname=fred
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14
```

```
POST /cgi-bin/program.pl HTTP/1.0
Accept: www/source
Accept: text/html
User-Agent: Lynx/2.4 libwww/2.14
Content-type: application/x-www-form-urlencoded
Content-length: 13
yourname=fred
```

```
<input type="text" name="yourname" id="yourid" />
```

</nerdy-stuff>

What does that mean?

- GET has an upper limit of the number of bytes of parameters and values (think about 2K)
- GET is used when your are reading or searching things
- POST is used when data is being created or modified
- Web search spiders will follow GET URLs but generally not POST URLs
- GET Urls should be “idempotent” - the same URL should give the “same thing” each time you access it

Fieldset and Legend

```
<form method="get" action="simpleform.html">
<fieldset>
<legend>All About You</legend>
 <p>
 <label for="yourname">Enter your name:</label>
 <input type="text" name="yourname" id="yourname" />
 </p>
 <p><input type="submit" /></p>
</fieldset>
</form>
```

Input Types

- Text
- Password
- Checkbox
- Radio Button
- Hidden
- Submit
- File

Text Input

```
<p>
 <label for="nameinp">Enter your name:</label>
 <input type="text" name="yourname" id="nameinp" />
</p>
<p>
 <label for="nickinp">Enter your nickname:</label>
 <input type="text" name="nickname" id="nickinp" value="Bob" />
</p>
```

The id= attribute is used to reference the field inside the HTML document. The name= attribute is the parameter name used to submit the data to the server.

Text fields can either start out blank of have content pre-populated.

Password Input Type

```
<p>
 <label for="password">Your password:</label>
 <input type="password" id="password" name="password" />
</p>
```

This only hides the password from view on the screen - to protect the password while in-transit, you need to send the data over https.

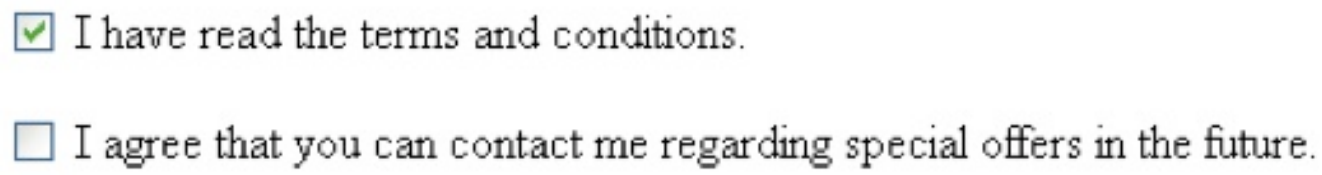
Hidden

- Hidden fields are used generally so that a program in a web server can send some internal information back to itself.

```
<input type="hidden" name="peekaboo" value="hereiam" />
```

Checkbox - Multiple Select

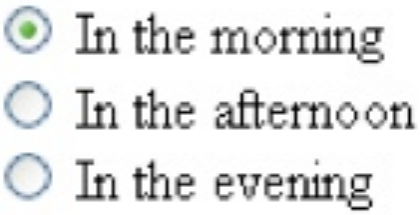
```
<p>
 <input type="checkbox" name="terms" id="termid" />
 <label for="termid">I have read the terms and conditions.</label>
</p>
<p>
 <input type="checkbox" name="offers" id="offerid" />
 <label for="offerid">I agree that you can contact me regarding
    special offers in the future.</label>
</p>
```



Checkbox - Preselected

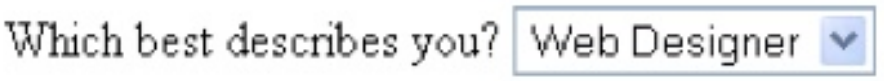
```
<p>
 <input type="checkbox" name="terms" id="termid" />
 <label for="termid">I have read the terms and conditions</label>
</p>
<p>
 <input type="checkbox" name="offers" id="offerid" checked="checked" />
 <label for="offerid">I agree that you can contact me regarding
special offers in the future</label>
</p>
```

Radio Buttons - Choice




```
<p>
 <input type="radio" name="timeslot" id="morning" value="morning" checked="checked" />
 <label for="morning">In the morning</label>
 <br />
 <input type="radio" name="timeslot" id="afternoon"  value="afternoon" />
 <label for="afternoon">In the afternoon</label>
 <br />
 <input type="radio" name="timeslot" id="evening" value="evening" />
 <label for="evening ">In the evening</label>
</p>
```

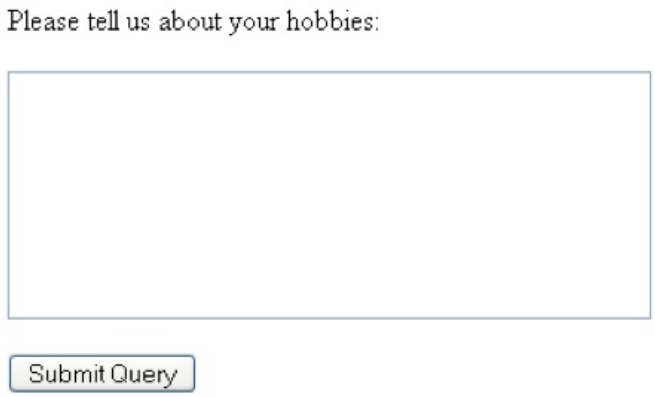
Drop Down List




```
<p>
 <label for="role">Which best describes you?</label>
 <select name="role" id="role">
 <option>Secretary</option>
 <option selected="selected">Web Designer</option>
 <option>Manager</option>
 <option>Cleaner</option>
 <option>Other</option>
 </select>
</p>
```

A drop-down list generates a single value when it is sent to the server.

Textarea for paragraphs




```
<p>
<label for="hobbies">Please tell us about your hobbies:</label>
</p>
<p>
<textarea name="hobbies" rows="7" cols="40" id="hobbies">
</textarea>
</p>
```

Textareas can become rich text areas - http://tinymce.moxiecode.com/

File Uploads

<input type=”file” name=”datain" accept="text/html">

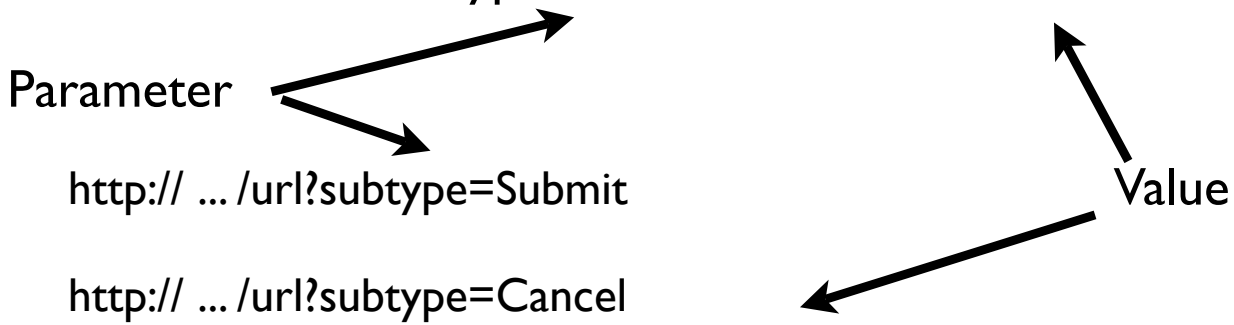
- File input is simple on the browser
- You can optionally insist on only certain file types
- File input processing depends on which software is receiving the file input on the server

Submit Button(s)

When you have multiple submit buttons the value can be used to figure out which button was pressed.

```
<p><input type="submit"/></p>

<!-- Multiple submit buttons -->
<p>
  <input type="submit" name="subtype" value="Submit"/>
  <input type="submit" name="subtype" value="Cancel"/>
</p>
```



Summary

- Forms allow us to layout areas for input and specify where and how to submit the input for processing on a web server
- GET is used for reading or querying information
- POST is used for modifying information